METHOD AND APPARATUS FOR SYNCHRONOUS PROJECT COLLABORATION

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/369,711, filed Apr. 2, 2002.

FIELD OF THE INVENTION

[0002] The present invention relates generally to project management systems and, more particularly, to protect management systems that facilitate the synchronous interaction of a number of individuals to create and modify documents and to perform other project tasks.

BACKGROUND OF THE INVENTION

[0003] Project management systems increase productivity and efficiency of members of a project team by automating the flow of information, including documents and files, among team members. Project management systems are often deployed to support collaborative work among a group of individuals, such as the members of a project team. Asynchronous collaboration systems allow team members to collaborate on one or more project tasks independently in time or space. Synchronous collaboration systems, on the other hand, allow team members to simultaneously collaborate on one or more project tasks in the same or a different location.

[0004] As the employees of an enterprise become more distributed in time and place, for example, due to flexible work hours, globalization and the distribution of enterprise employees to avoid the destruction of a centralized enterprise location, it becomes even more important to provide team members with an effective tool for asynchronous and synchronous collaboration. In today's enterprise environment, it is important for a project management system to permit distributed team members to initiate ad-hoc virtual meetings, for example, over the Internet. Generally, such project management systems must allow distributed team members to communicate and interact as if the team members were in the same place.

[0005] When team members collaborate, they often share and revise documents, such as tables, charts and drawings. Often, the various requested revisions from team members on a particular document may cause a conflict. For instance, one team member may initiate a command to move a particular object to the left, while another team member may initiate a command to move the same object to the right. Even when such conflicting commands occur close in time, however, the team members should see the document in the same way as the results of all of the changes made from the entire team.

[0006] Most document management systems prevent conflicting changes from multiple team members by employing a "token." One or more tokens are associated with each shared document. If a team member desires to make a revision to a document, the team member must first obtain the appropriate token(s). Once the team member has obtained the token(s) and made the desired revisions, the token should be released and returned to a token pool. If one team member has the token, then all other team members must wait to make any further revisions to the associated document (or document portion). In this manner, the document management system can safely serialize revisions and ensure that different team members do not make conflicting revisions to shared documents.

[0007] Such token-based mechanisms, however, introduce a delay before a team member can make a revision, as the team member must first obtain the token before performing most actions on the shared document. This is especially true when the token is stored at a central server, which is often the case. In addition, when one team member has possession of the token, all other team members are unable to manipulate the document. Finally, if the computer of the team member currently with possession of the token happens to crash, then the entire system is locked-up for at least a minimum time-out period.

[0008] A need therefore exists for an improved project management system and method that facilitate the synchronous and asynchronous interaction of a number of individuals to create and modify documents and other project tasks. A need also exists for an improved project management system and method that incrementally provides a synchronous collaboration system to extend a network asynchronous collaboration system so that one or more users may transition between asynchronous and synchronous collaboration modes. A further need exists for a mechanism that determines a canonical ordering of conflicting change requests in a shared document without first requiring the user to obtain token. Yet another need exists for a method and apparatus for presenting shared documents to each team member in the same way at any given time.

SUMMARY OF THE INVENTION

[0009] The present invention provides a project management system that allows one or more team members to work on a project. Generally, a method and apparatus are provided for peer-to-peer sharing of documents in asynchronous and synchronous collaboration modes. The present invention allows documents to be revised by individual team members in an asynchronous collaboration mode or as the result of group meetings (in or more locations) by multiple team members in a synchronous collaboration mode. According to one aspect of the invention, a synchronous collaboration system is provided as an incremental addition that extends a conventional asynchronous collaboration system. In this manner, the present invention allows one or more users to easily transition between asynchronous and synchronous collaboration modes.

[0010] According to another aspect of the present invention, a plurality of users can interact in a synchronous collaboration mode to create and modify documents and perform other project tasks without requiring a token. Each user can submit potentially conflicting change requests for an object spontaneously and concurrently. For example, a first user might request that an object is moved to the left while another user might request that the same object is moved to the right. A serializer initially receives each of the change requests and serializes them, for example, based on an arrival time or a global time stamp. The serialized requests are then sent in order to a broadcaster that broadcasts the requests to all users. For example, the change requests can be broadcast to all currently active users in